

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-216008

(43)Date of publication of application : 10.08.2001

(51)Int.Cl. G05B 19/404 B25J 9/10 G05B 19/19
G05B 19/416 G05D 3/12

(21)Application number : 2000-027768

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 04.02.2000

(72)Inventor : MAEKAWA SEISEKI

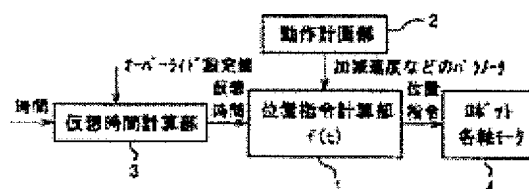
MAKITA HIROYUKI

(54) ROBOT CONTROL METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a robot control method for simplifying the teaching of a robot by generating the same target locus at the time of over-ride change as that at the time of over-ride 100% even at the time of connecting locus.

SOLUTION: A virtual time is introduced, and the virtual time is updated in each position command generation cycle. Then, the change rate of the virtual time is changed according to the over-ride value, and the generation of the position command is operated based on not a real time but the virtual time.



対応なし、英抄

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-216008

(P2001-216008A)

(43) 公開日 平成13年8月10日 (2001.8.10)

(51) Int.Cl. ⁷	識別記号	FI	テーマコード* (参考)
G 0 5 B 19/404		G 0 5 B 19/404	L 3 F 0 5 9
B 2 5 J 9/10		B 2 5 J 9/10	A 5 H 2 6 9
G 0 5 B 19/19		G 0 5 B 19/19	L 5 H 3 0 3
19/416		19/416	L 9 A 0 0 1
G 0 5 D 3/12		G 0 5 D 3/12	S

審査請求 未請求 請求項の数 9 OL (全 10 頁) 最終頁に続く

(21) 出願番号 特願2000-27768 (P2000-27768)

(22) 出願日 平成12年2月4日 (2000.2.4)

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 前川 清石

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(72) 発明者 牧田 裕行

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(74) 代理人 100102439

弁理士 宮田 金雄 (外1名)

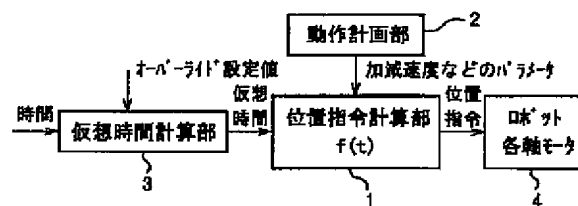
最終頁に続く

(54) 【発明の名称】 ロボット制御方法

(57) 【要約】 (修正有)

【課題】 軌跡接続時においても、オーバーライド変更時もオーバーライド100%時と同一の目標軌道を生成できるようにし、ロボットの教示を容易にするロボット制御方法を得る。

【解決手段】 仮想的な時間を導入し、位置指令生成周期毎にこの仮想的時間を更新してゆく。そしてオーバーライドの値に応じて仮想的な時間の変化率を変更し、位置指令の生成を実時間ではなく仮想的な時間に基づいて行う。



【特許請求の範囲】

【請求項1】 指定されたオーバーライドに基づいて仮想的な時間の変化率を変更し、仮想的な時間に基づいて位置指令の生成を行うことを特徴とするロボット制御方法。

【請求項2】 指定されたオーバーライドに基づいて仮想的な時間の変化率を変更し、仮想的な時間が位置指令生成周期を越えたときのみ台形速度指令生成を行うことを特徴とするロボット制御方法。

【請求項3】 指定されたオーバーライドを制御装置内部に記憶し、指定されたオーバーライドに滑らかに追従するように逐次実オーバーライドを変更し、実オーバーライドに基づいて仮想的な時間の変化率を変更することを特徴とする請求項1または請求項2記載のロボット制御方法。

【請求項4】 指定されたオーバーライドもしくは実オーバーライドで規制される速度以下となるように仮想的な時間の変化率を逐次変更することを特徴とする請求項1または請求項2記載のロボット制御方法。

【請求項5】 ロボットの一時停止時には仮想的な時間の変化率を0まで滑らかに減少させることによりロボットを停止させることを特徴とする請求項1または請求項2記載のロボット制御方法。

【請求項6】 ロボットの一時停止後の再開時には仮想的な時間の変化率を0から滑らかに増加させることを特徴とする請求項5記載のロボット制御方法。

【請求項7】 ロボットの緊急停止時に仮想的な時間の変化率を一定期間負の値にした後0にすることを特徴とする請求項1または請求項2記載のロボット制御方法。

【請求項8】 ロボットの現在の位置誤差を常に監視し、位置誤差が許容値以上となった場合に仮想的な時間の変化率を、指定されたオーバーライドもしくは実オーバーライドよりも小さくすることを特徴とする請求項1または請求項2記載のロボット制御方法。

【請求項9】 ロボットの手先位置を常に監視し、手先位置の目標軌道からのずれが許容値以上となった場合に仮想的な時間の変化率を、指定されたオーバーライドもしくは実オーバーライドよりも小さくすることを特徴とする請求項1または請求項2記載のロボット制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、モータを用いてロボットを駆動するロボット制御方法に関するものである。

【0002】

【従来の技術】ロボットなどの自動機械を教示点に沿って厳密に動かす必要のない場合は、第1の動作である動作1が終了する前に第2の動作である動作2を開始することにより、動作時間を短縮するとともに、ロボットを滑らかに動作させることが行われてきた。例えば、特開

(2)

特開2001-216008

2

昭64-26911号公報に記載されているように、動作1の減速開始と同時に動作2の加速を開始することにより、2つの動作を滑らかに接続する軌道を動作1と動作2との合成により生成する方法が用いられている。

【0003】一方、実際にロボットを使用する際には、ロボット動作プログラムを作成した後、作成したロボット動作プログラムをオーバーライドを下げて（プログラムはそのまま制御盤等から速度を下げて）実際に動作させるテスト動作を行い、ロボットの手先が通る軌道が適切かどうかを確認することが多く行われる。ロボットの手先が通る軌道が適切でない場合は、プログラムの修正を行い、再度テスト動作を行う。ロボットの手先が通る軌道が適切であると確認した後、オーバーライドを上げて目的とする作業をロボットに繰り返し行わせる。

【0004】また、オーバーライドを下げる場合は、指令速度をクランプする方法が一般に用いられていた。図16にオーバーライド100%及び50%時の速度指令を示す。

【0005】

【発明が解決しようとする課題】AからBへの直線動作である動作1とBからCへの直線動作である動作2を動作1の減速開始時に動作2を開始する場合のロボット手先の軌道を図17、速度指令を図18に示す。図17に示したように、上記従来技術では、オーバーライドを下げているときのロボット手先の通る軌道とオーバーライドを上げているときのロボット手先の通る軌道は2つの動作の合成部分でずれが生じていた。そのためテスト動作を行う際には、低いオーバーライドから始めて徐々にオーバーライドを上げながらテスト動作を繰り返す必要があり、テスト動作に多大な時間を要するという問題点があった。

【0006】また、動作1と動作2の合成中に停止命令が入ると、図19に示すようにロボット手先が本来の軌道からずれてしまうといった問題点があった。

【0007】この発明は上記のような問題点を解消するためになされたもので、軌跡接続時においても、オーバーライド変更時もオーバーライド100%時と同一の目標軌道を生成できるようにし、ロボットの教示を容易にすることができるとしてロボット制御方法を提供することを目的とする。

【0008】

【課題を解決するための手段】この発明の第1の構成によるロボット制御方法は、指定されたオーバーライドに基づいて仮想的な時間の変化率を変更し、仮想的な時間に基づいて位置指令の生成を行うものである。

【0009】この発明の第2の構成によるロボット制御方法は、指定されたオーバーライドに基づいて仮想的な時間の変化率を変更し、仮想的な時間が位置指令生成周期を越えたときのみ台形速度指令生成を行うものである。

50

【0010】この発明の第3の構成によるロボット制御方法は、指定されたオーバーライドを制御装置内部に記憶し、指定されたオーバーライドに滑らかに追従するように逐次実オーバーライドを変更し、実オーバーライドに基づいて仮想的な時間の変更率を変更するものである。

【0011】この発明の第4の構成によるロボット制御方法は、指定されたオーバーライドもしくは実オーバーライドで規制される速度以下となるように仮想的な時間の変更率を逐次変更するものである。

【0012】この発明の第5の構成によるロボット制御方法は、ロボットの一時停止時には仮想的な時間の変更率を0まで滑らかに減少させることによりロボットを停止させるものである。

【0013】この発明の第6の構成によるロボット制御方法は、ロボットの一時的停止後の再開時には仮想的な時間の変更率を0から滑らかに増加させるものである。

【0014】この発明の第7の構成によるロボット制御方法

$$kt(k) = kt(k-1) + dkt(k) \cdot \Delta t$$

ここで、kはk回目の位置指令生成周期を意味し、kt(k)はk回目の位置指令生成周期における仮想的な時間、dkt(k)はk回目の位置指令生成周期における仮想的な時間の変更率、 Δt は位置指令生成周期である。オーバーライドの値に応じて仮想的な時間の変更率を変更し、位置指令の生成を実時間ではなく仮想的な時間に基づいて行う。

【0019】ロボット動作中にオーバーライドが変更されても速やか、かつ、滑らかに適応するため、現在設定されているオーバーライドを目標オーバーライドとして制御装置内部に記憶するとともに実オーバーライドを目標オーバーライドに滑らかに追従させる。仮想的な時間の変更率を実オーバーライドに基づいて変更することにより、ロボット動作中にオーバーライドが変更されても速やか、かつ、滑らかに適応できる。

【0020】ロボットの一時的停止時には仮想的な時間の変更率を滑らかに0まで減少させて停止する。一時停止後の動作再開時には仮想的な時間の変更率を滑らかに0から増加させる。

【0021】ロボットが障害物に衝突したことを検知し停止させる場合は、仮想的な時間の変更率を減少させて一定期間負の値に設定した後、逆に0まで増加させてか※

$$kt(k) = kt(k-1) + dkt(k) \cdot \Delta t$$

ここで、kはk回目の位置指令生成周期を意味し、kt(k)はk回目の位置指令生成周期における仮想的な時間、dkt(k)はk回目の位置指令生成周期における仮想的な時間の★

$$dkt(k) = x/100$$

とし、動作中にオーバーライドが変更された場合は、動作終了後dktの値を変更したオーバーライド値から

(3)式に基づいて変更する。

【0025】位置指令計算部1では、実時間 $t=k \cdot \Delta t$ で

＊方法は、ロボットの緊急停止時に仮想的な時間の変更率を一定期間負の値にした後0にするものである。

【0015】この発明の第8の構成によるロボット制御方法は、ロボットの現在の位置誤差を常に監視し、位置誤差が許容値以上となった場合に仮想的な時間の変更率を、指定されたオーバーライドもしくは実オーバーライドよりも小さくするものである。

【0016】この発明の第9の構成によるロボット制御方法は、ロボットの手先位置を常に監視し、手先位置の目標軌道からのずれが許容値以上となった場合に仮想的な時間の変更率を、指定されたオーバーライドもしくは実オーバーライドよりも小さくするものである。

【0017】

【発明の実施の形態】この発明の基本的事項はつぎの通りであり、具体的には実施の形態1～9において図を用いて説明する。

【0018】仮想的な時間を導入し、位置指令生成周期毎に(1)式のように仮想的な時間を更新していく。

$$(1)$$

20※ら停止する。

【0022】作用の概略はつぎの通りである。変化率をオーバーライドに応じて変更する仮想的な時間に基づいて位置指令を生成するため、オーバーライドによらずロボット手先の軌道は一定となり、テスト動作に要する時間を短縮できる。また、一時停止／再開時に仮想的な時間の変更率を増減することによって減速停止／再加速を実現しているため、一時停止／再開時にも本来の軌道からそれることがない。また、ロボットが障害物に衝突した場合等にロボットを停止させる際、軌道上で衝突以前の地点へ自動的に戻すことが可能になる。

【0023】実施の形態1. 本実施の形態における指令値生成部を図1に示す。まず、時間を入力とし、加減速、最高速度等をパラメータとして位置指令生成周期毎の位置指令を計算するための関数 $f(t)$ を位置指令計算部1に用意する。ロボットにプログラム動作させる際には、まず、動作計画部2で当該動作の加減速、最高速度等のパラメータを決定し、位置指令計算関数 $f(t)$ のパラメータ値を決定したパラメータに変更する。

【0024】仮想時間計算部3では、位置指令生成周期毎に(2)式のように仮想的な時間を更新していく。

$$(2)$$

★変化率、 Δt は位置指令生成周期である。オーバーライドがx%の時は

$$(3)$$

はなく、仮想時間計算部3で計算した仮想時間 $kt(k)$ を入力として生成周期毎の位置指令を計算する。例えば指定したオーバーライドが50%の場合、仮想時間計算部3で計算される仮想時間は実時間の50%の値となるため、

最高速度はオーバーライドが100%の時と比べると50%になる。

【0026】2つ以上の動作を合成する際には、合成される各動作位置指令計算関数 $f(t)$ の入力を同じ仮想時間 kt とする。実施の形態1でA点からB点への直線動作である動作1が終了する前にB点からC点への直線動作である動作2を開始し、動作2開始後は動作1と動作2の合成で目標軌道を生成した場合のロボットの手先が描く軌道を図2に示す。また、動作1及び動作2の速度指令を図3に示す。以上のように求められた位置指令により

$$dkt(k) = rov(k)/100$$

で求め、(4)式で求めた $dkt(k)$ を用いて、(2)式で逐次仮想時間 $kt(k)$ を計算していく。仮想時間計算部3で計算した仮想時間を入力として位置指令計算部1で位置指令を計算する部分は実施の形態1と全く同一である。

【0028】実施の形態3. 本実施の形態の指令値生成部を図5に示す。実施の形態2とは位置指令計算部1が異なる。図5において、速度台形に基づく位置指令生成部1-1とは、速度パターンが台形パターンとなる位置指令を生成する部分で、位置指令を滑らかにするため、図5の移動平均フィルタ相当部を設けている。速度台形に基づく位置指令生成部1-1では仮想時間が位置指令生成周期を越える度に、速度台形に基づく位置指令の計算を行う。ここで、仮想時間と位置指令生成周期刻みが図6の関係にある時を例に挙げて説明する。図6におい※

$$nd = nd1 + nd2$$

$$mnd = m1*nd1 + m2*nd2$$

とする。このとき、移動平均フィルタ相当部の時刻 $n*\Delta t$ における出力を $pcmd(n)$ とすると、図6の $n=k$ の場合の★

$$\begin{aligned} pcmd(n) = & pcmd(n-1) + b*(m1*(p0(j) - p0(j-nd1)) \\ & + m2*(p0(j-nd1) - p0(j-nd)))/mnd \\ & + a*(m1*(p0(j-1) - p0(j-1-nd1)) \\ & + m2*(p0(j-1-nd1) - p0(j-1-nd)))/mnd \end{aligned} \quad (7)$$

図6の $n=k+1$ の場合のように仮想時間が位置指令生成周期を越えない場合は

$$\begin{aligned} pcmd(n) = & pcmd(n-1) + c*(m1*(p0(j) - p0(j-nd1)) \\ & + m2*(p0(j-nd1) - p0(j-nd)))/mnd \end{aligned} \quad (8)$$

で計算される。ここで、 a 、 b 、 c は図6に記載の定数で、

$$a + b = dkt(k-1) \quad (9)$$

$$c = dtk(k) \quad (10)$$

である。

【0030】実施の形態4. 本実施の形態の指令値生成部を図8に示す。図8において、実オーバーライド計算部5、動作計画部2、速度台形に基づく位置指令生成部1-1、移動平均フィルタ相当部1-2は図5における、実オーバーライド計算部、動作計画部、速度台形に☆

$$abs(vcmd) > vmax*(rov(k))^{\alpha} \text{の時}$$

$$dkt(k) = (rov(k))^{\alpha / (\alpha - 1)} * (vmax/abs(vcmd))^{1 / (\alpha - 1)} \quad (11)$$

上記以外の時

※【0027】実施の形態2. 本実施の形態における指令値生成部を図4に示す。本実施の形態と実施の形態1の大きな差異は、実オーバーライド計算部5を備えることである。実オーバーライド計算部5には2段の移動平均フィルタを備えている。1段目の移動平均フィルタへの入力は現在のオーバーライドの設定値 $ov(k)$ で、2段目の移動平均フィルタの入力は1段目の移動平均フィルタの出力である。2段目の移動平均フィルタの出力を実オーバーライド $rov(k)$ として、仮想時間計算部3に出力する。ここで、 k は k 番目の位置指令生成周期を意味する。仮想時間計算部3では仮想時間変化率 $dkt(k)$ を

$$(4)$$

※で、 Δt は位置指令生成周期、 j は j 番目の位置指令生成周期刻みを意味する。図6では実時間が $\Delta t * k$ の時に、仮想時間は j 番目の位置指令生成周期刻みを越える。従って、実時間が $\Delta t * k$ の時に、 j 番目の位置指令生成周期刻みにおける速度台形に基づく位置指令 $p0(j)$ を計算する。実時間が $\Delta t * (k+1)$ の時、仮想時間は位置指令生成周期刻みを越えないので、速度台形に基づく位置指令は計算しない。

【0029】次に移動平均フィルタ相当部1-2での処理について説明する。位置指令を滑らかにするために用いる移動平均フィルタとしては、図7に示すような前半部と後半部で重みを変更できるフィルタとする。図7において $m1$ は前半部の重み、 $m2$ は後半部の重み、 $nd1$ は前半部のサンプル数、 $nd2$ は後半部のサンプル数とし、 n 、 d 、 mnd をそれぞれ

$$(5)$$

$$(6)$$

★ように仮想時間が j 番目の位置指令生成周期刻みを越える場合は

$$dtk(k) = 1$$

とする。ここで、 v_{max} は動作計画部2から仮想時間計算部3に入力される最高速度である。各動作において動作開始後しばらくはオーバーライド100%の時と全く同じ挙動となり、速度指令が（プログラムで指定された）最高速度に（実オーバーライド/100）の α 乗を掛けた値より大きくなった時点から仮想時間が徐々に遅くなっていくため、速度が上がりにくくなっていき、オーバーライドで指定した速度に収束する。減速を始めると仮想時間の変化率は徐々に増加し、速度指令が（プログラムで指定された）最高速度に（実オーバーライド/100）の α 乗を掛けた値より小さくなった時点で1に戻る。2つ以上の動作を合成している区間では、それぞれの動作に対して（11）、（12）式で仮想時間の変化率を計算し、最小値を採用する。

【0031】実施の形態5、制御装置に付属のオペレータパネル等から停止ボタンが押され、停止信号が発せられた場合、停止信号を実施の形態3に記載の速度台形に基づく位置指令生成部に入力し、速度台形に基づく位置指令生成部において減速停止パターンを生成するとともに、停止信号を実オーバーライド計算部に入力し、仮想時間の変化率を滑らかに0まで減少させる。仮想的な時間の変化率を0にする方法としては、例えば実施の形態3の制御装置において停止ボタンが押され、停止信号が実オーバーライド計算部に入力された後は、実オーバーライド計算部において、現在のオーバーライド設定値 $ov(k)$ の値を0にする方法が挙げられる。実施の形態5の指令値生成部を図9に示す。

【0032】実施の形態6、制御装置に付属のオペレータパネル等から停止ボタンが押された場合、停止信号が実オーバーライド計算部5に入力され、仮想時間の変化率を滑らかに0まで減少させる。仮想時間の変化率が0になれば、仮想時間が進まないため、位置指令計算部1で出力される位置指令は同一の値となるため、ロボットはオーバーライド100%の時に通る軌道上で停止する。仮想的な時間の変化率を0にする方法としては、例えば実施の形態3の制御装置において停止ボタンが押されて停止信号が実オーバーライド計算部5に入力されてから、再開ボタンが押されて再開信号が実オーバーライド計算部5に入力されるまで、実オーバーライド計算部5において、現在のオーバーライドの設定値 $ov(k)$ の値を0にする方法が挙げられる。停止中は仮想的な時間が進まないだけで、位置指令生成部では仮想時間が進むのを待ち続けるため、位置指令生成部の情報は停止時のまま保持され続ける。また、再開時には仮想時間の変化率を0から徐々に増加することにより、ロボットを加速する。例えば、実施の形態3の制御装置において、再開動作の際に、再開信号が実オーバーライド計算部5に入力されれば、実オーバーライド計算部5における現在のオーバーライド設定値 $ov(k)$ の値を元に戻せばよい。実施の形態

6の指令値生成部を図10に示す。また、実施の形態6で、AからBへの直線動作である動作1とBからCへの直線動作である動作2を合成している途中で停止ボタンを押したときのロボットの軌道を図11に示す。

【0033】実施の形態7、実施の形態7による制御装置の位置指令生成部を図12に示す。本実施の形態では、実施の形態1と同様に時間を入力とし、加減速、最高速度等をパラメータとして位置指令生成周期毎の位置指令を計算するための関数 $f(t)$ を位置指令計算部1に用意する。ロボットにプログラム動作させる際には、まず、動作計画部2で当該動作の加減速、最高速度等のパラメータを決定し、位置指令計算関数 $f(t)$ のパラメータ値を決定したパラメータに変更する。実オーバーライド計算部5には実施の形態2と同様に2段の移動平均フィルタを備えている。1段目の移動平均フィルタへの入力には現在のオーバーライドの設定値 $ov(k)$ で、2段目の移動平均フィルタの入力は1段目の移動平均フィルタの出力である。2段目の移動平均フィルタの出力を実オーバーライド $rov(k)$ として、仮想時間計算部3に出力する。ここで、 k は k 番目の位置指令生成周期を意味する。仮想時間計算部3では仮想時間変化率 $dkt(k)$ を（4）式で求め、（4）式で求めた $dkt(k)$ を用いて、（2）式で逐次仮想時間 $kt(k)$ を計算していく。仮想時間計算部3で計算した仮想時間を入力として位置指令計算部1で位置指令を計算する部分は実施の形態1、2と全く同一である。

【0034】モータ変位、モータトルクから衝突を検出する衝突検知部で衝突が検知されると、検知信号が実オーバーライド計算部5に出力される。実オーバーライド計算部5に検知信号が入力されると、実オーバーライド計算部5において、現在のオーバーライドの設定値 $ov(k)$ があらかじめパラメータで指定された負の値 cov に設定されるとともに、衝突検知時の仮想時間 $kt(k)$ が記憶される。（記憶された仮想時間の値を ckt とする。）実オーバーライド計算部5では衝突検知後オーバーライド設定値が $ov(k) = cov$ に変更されたとして、2段の移動平均フィルタの処理を行う。仮想時間が $kt(k)$ が $ckt + ct$ 以下となれば、オーバーライドの設定値 $ov(k)$ を0にする。ここで、 ct もあらかじめ設定するパラメータ値である。上記のような構成にすることにより、衝突検知時には軌道上を衝突検知以前の地点まで戻って停止することが可能になる。そのため、別の障害物に衝突する、ロボットに過大な力が作用したまま停止するなどの問題を防止できる。

【0035】実施の形態8、実施の形態8による制御装置の位置指令生成部を図14に示す。ロボットを駆動するモータの速度トルク特性は、一般に図13のようになる。そのため、図13の v_1 以上の速度領域でロボットを動作させる際には、トルク飽和が発生し、位置誤差が

過大となり、ロボットがエラー停止する場合もある。そこで、実施の形態8では、位置誤差のしきい値を設定し、位置誤差がしきい値以上であり、かつ、その軸の速度指令が v_1 以上の場合に仮想時間の変化率を減少させる。具体的には、速度指令と位置誤差をしきい値判定部7に入力し、速度指令が v_1 以上で、かつ、位置誤差がしきい値以上の場合、しきい値判定信号を出力する。しきい値判定信号が出力されている間は、実オーバーライド計算部で、オーバーライドの設定値を「現在のオペレータパネルなどでの設定値*パラメータであらかじめ決められた減少率」に設定する。位置誤差がしきい値以下になった場合、もしくは速度指令が v_1 以下になった場合は、しきい値判定部からしきい値判定信号が出力されなくなる。しきい値判定信号が出力されなくなると、実オーバーライド計算部において、オーバーライドの設定値を、現在のオペレータパネルなどでの設定値に戻す。

【0036】実施の形態9. ロボットの手先に直線移動動作を行わせる際、まず、手先位置計算部8に各軸の現在のモータ変位、モータ速度を入力し、現在の手先位置を計算する。ロボットを剛体と見なす場合は、モータ変位から順変換して手先位置を計算する。ロボットをリンク部は剛体であるが、関節部は線形バネで構成されていると見なす場合は、入力したモータ速度からモータ加速度を計算する。次に、モータ加速度、モータ速度、モータ変位から、各軸の線形バネに作用するトルクをロボットの運動方程式に基づいて計算する。計算したトルクに基づいて弾性変形分を補正した関節変位を求めてから順変換して手先位置を計算する。

【0037】実施の形態9による制御装置の位置指令生成部を図15に示す。ロボットが、現在加速区間、等速区間、減速区間のいずれにあるかを判別する区間判別信号が、位置指令計算部1から手先位置計算部8に入力される。ロボットが加速区間、もしくは等速区間にあるときは、手先位置計算部8で計算した手先位置と目標軌道である直線との距離を求め、直線との距離が許容値以上の時は、許容値超過判別信号を出力し、実オーバーライド計算部5で、オーバーライドの設定値を（現在のオペレータパネルなどでの設定値*パラメータであらかじめ決められた減少率）に設定する。直線との距離が許容値以下になれば、許容値超過判別信号の出力を停止し、実オーバーライド計算部で、オーバーライドの設定値を現在のオペレータパネルなどでの設定値に戻す。

【0038】

【発明の効果】この発明の第1および第2の構成であるロボット制御方法によれば、オペレータパネルなどで設定するオーバーライドを変更しても、ロボット手先が通る軌道は常に一定となる。したがって、テスト動作に要する時間を短縮できる効果がある。

【0039】この発明の第3の構成であるロボット制御

方法によれば、請求項3の発明は、ロボット動作中にオーバーライドを変更しても、変更したオーバーライドに滑らかに移行できる効果がある。

【0040】この発明の第4の構成であるロボット制御方法によれば、動作開始後しばらくと動作終了前のしばらくの期間はオーバーライド100%時と全く同一の動作が行えるため、オーバーライドを下げているときでも、必要以上に動作時間が長くなることを防止する効果がある。

【0041】この発明の第5および第6の構成であるロボット制御方法によれば、仮想的な時間の変化率を増減することにより停止・再開動作が行えるため、停止・再開動作においても目標軌道からのずれが発生しない効果がある。

【0042】この発明の第7の構成であるロボット制御方法によれば、ロボットの衝突を検知して停止する際には、軌道からずれることなく衝突以前の地点で停止することができ効果がある。

【0043】この発明の第8の構成であるロボット制御方法によれば、各軸の位置誤差がしきい値以上になったときに自動的に速度を低下させるため、位置誤差過大エラーによる停止を防止できる効果がある。

【0044】この発明の第9の構成であるロボット制御方法によれば、目標軌道からのずれが許容値以上となったときに自動的に速度を下げるのが可能なため、目標軌道からのずれを許容値以下に抑えながら高速化を図ることが容易に実現できる効果がある。

【図面の簡単な説明】

【図1】 実施の形態1の位置指令生成部を示す図である。

【図2】 実施の形態1における手先の軌道を示す図である。

【図3】 実施の形態1における速度指令を示す図である。

【図4】 実施の形態2の位置指令生成部を示す図である。

【図5】 実施の形態3の位置指令生成部を示す図である。

【図6】 仮想時間と指令値生成周期刻みの関係を示す図である。

【図7】 移動平均フィルタを示す図である。

【図8】 実施の形態4の位置指令生成部を示す図である。

【図9】 実施の形態5の位置指令生成部を示す図である。

【図10】 実施の形態6の位置指令生成部を示す図である。

【図11】 実施の形態6における停止ボタン入力時の手先の軌道を示す図である。

【図12】 実施の形態7の位置指令生成部を示す図で

ある。

【図13】 モータの速度-トルク特性を示す図である。

【図14】 実施の形態8の位置指令生成部を示す図である。

【図15】 実施の形態9の位置指令生成部を示す図である。

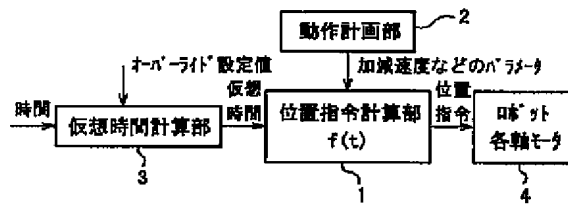
【図16】 オーバーライドを下げた時の従来の速度指令を示す図である。

【図17】 従来方式における手先の軌道を示す図である。

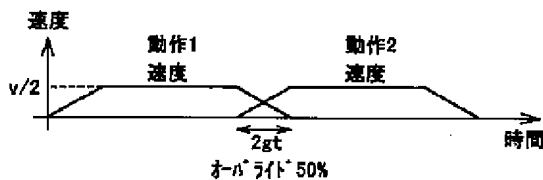
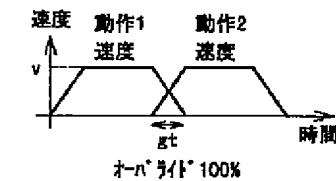
10

*

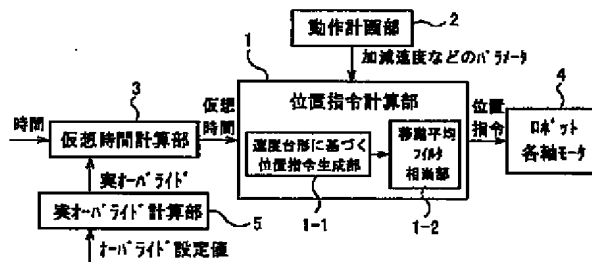
【図1】



【図3】



【図5】



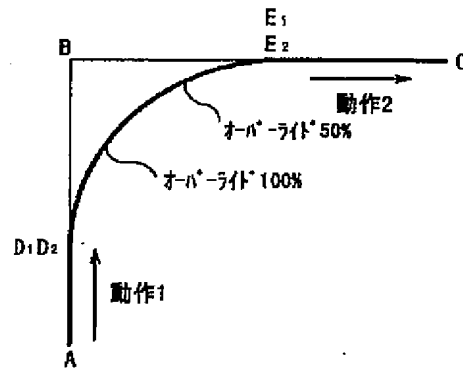
*【図18】 従来方式における速度指令を示す図である。

【図19】 従来方式における停止ボタン入力時の手先の軌道を示す図である。

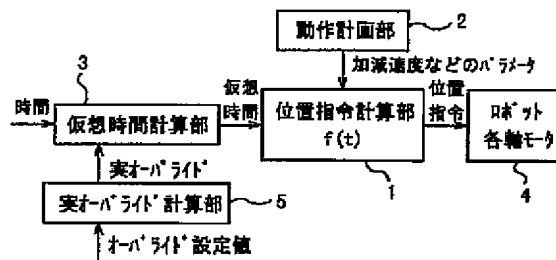
【符号の説明】

1 位置指令計算部、1-1 位置指令生成部、1-2 移動平均フィルタ相当部、1-3 差分部、2 動作計画部、3 仮想時間計算部、4 ロボット各軸モータ、5 実オーバーライド計算部、6 衝突検知部、7 しきい値判定部、8 手先位置計算部。

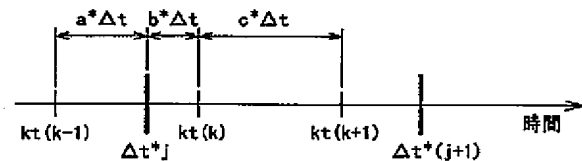
【図2】



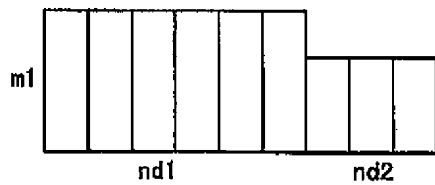
【図4】



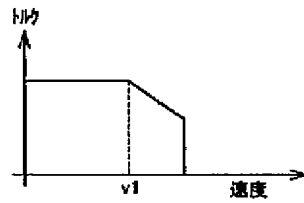
【図6】



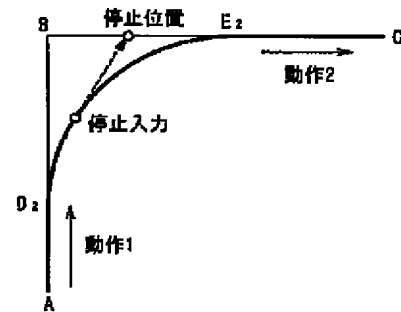
【図7】



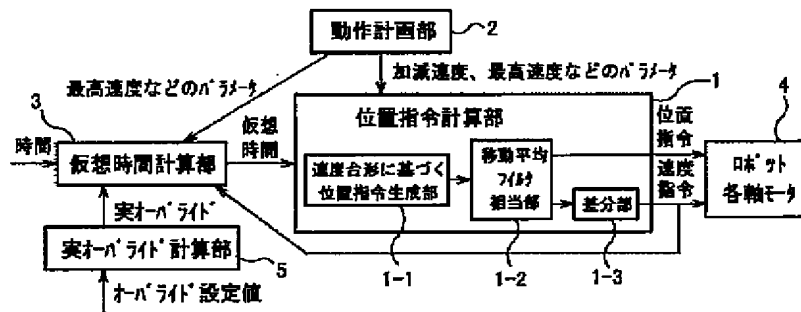
【図13】



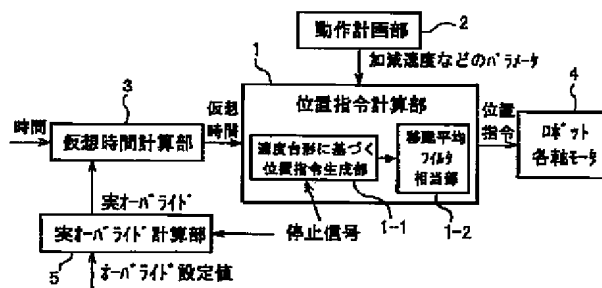
【図19】



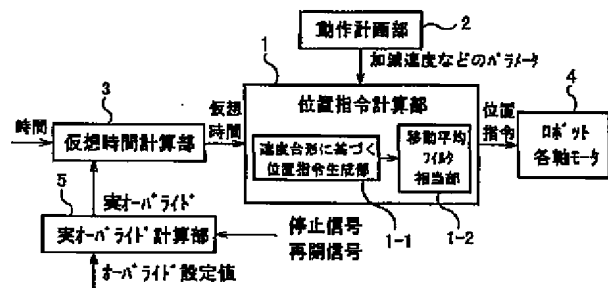
【図8】



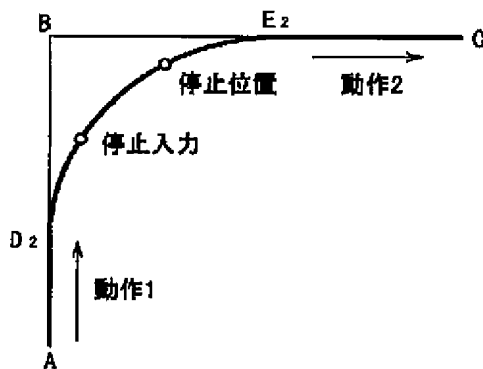
【図9】



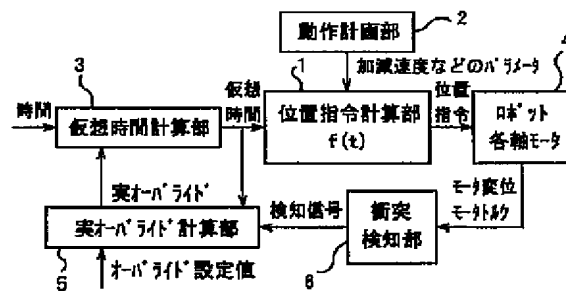
【図10】



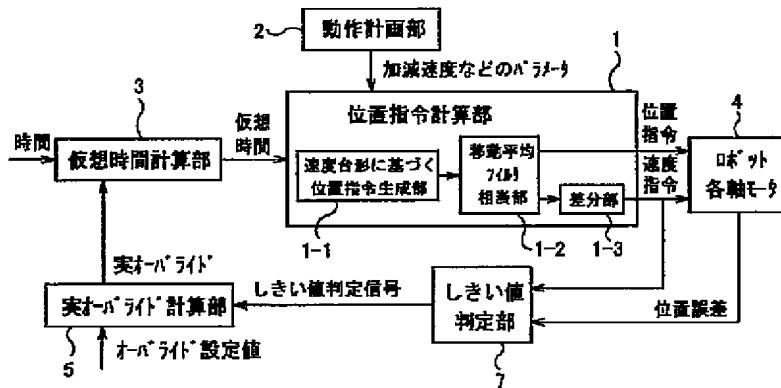
【図11】



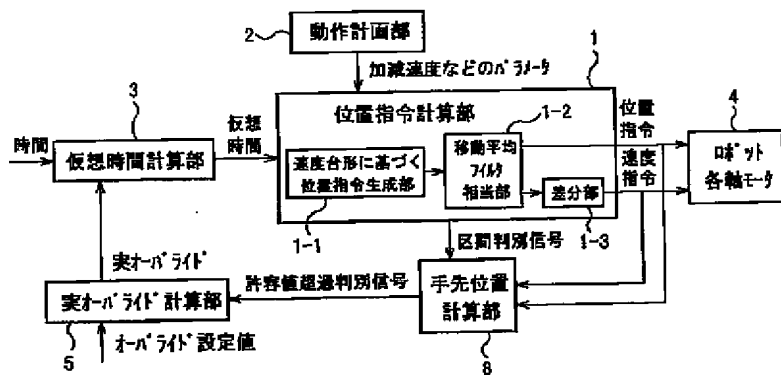
【図12】



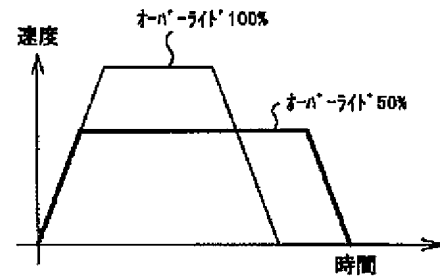
【図14】



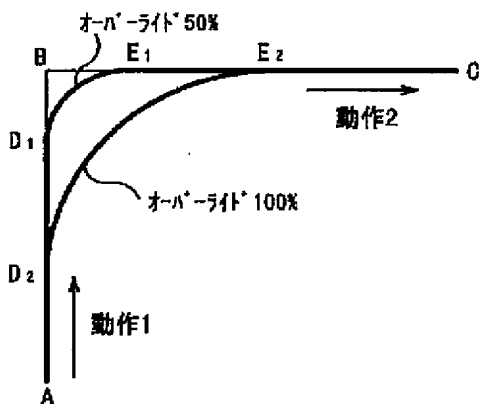
【図15】



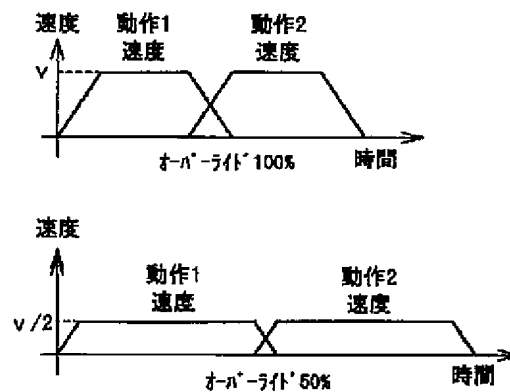
【図16】



【図17】



【図18】



フロントページの続き

(51)Int.Cl. ⁷	識別記号	F I	テーマコード (参考)
G 0 5 D 3/12	3 0 6	G 0 5 D 3/12	3 0 6 R

F ターム (参考) 3F059 FA07 FB05 FC02 FC09
 5H269 AB33 BB09 CC09 EE01 EE16
 GG02 NN01 NN07 QC10 SA01
 5H303 AA10 BB02 BB07 CC02 DD01
 EE03 EE07 HH05 KK14 KK17
 9A001 HH19 HH34 KK32